Intel® Telco Alarms Manager

User's Guide

Document Release Date: March 2003

Legal Information

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in HP's Terms and Conditions of Sale for such products, HP assumes no liability whatsoever, and HP disclaims any express or implied warranty, relating to sale and/or use of HP products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. HP products are not intended for use in medical, life saving, or life sustaining applications. HP may make changes to specifications and product descriptions at any time, without notice.

Copyright © Hewlett-Packard Corporation 2002.

* Other brands and names may be claimed as the property of others.

Table of Contents

List of Tables

List of Figures

Introduction

The Telco Alarms Manager (TAM) is a telecom server software component designed to manage the Telco Alarms Panel located on the front panel of Intel® Carrier Grade Servers.

The Telco Alarms Panel is a set of three LEDs that are labeled to indicate a level of severity of any current system alarms: "CRT" for "Critical", "MJR" for "Major", and "MNR" for "MINOR". A fourth LED labeled "PWR" indicates that at least one of the current system alarms is related to a power subsystem condition. Traditional public switched telephone network (PSTN) equipment designs have established a de facto standard for the behavior of the Telco Alarms Panel. The panel LED that corresponds to the most severe active alarm condition is asserted along with the PWR LED, if any active alarm concerns power. As alarm conditions are cleared, the state of the panel is reassessed and, if necessary, modified to reflect the resulting most-severe alarm condition.

The Telco Alarms Manager software is designed to provide alarms panel state management for any number of alarm generating components in the system. The TAM receives alarm state requests from software agents and consolidates those requests to determine the proper alarm panel LED status. As the software agents send alarm clear requests, the TAM clears those conditions and reassesses and reasserts the alarms panel LEDs as needed.

The TAM also includes pre-built event forwarders that work work with Intel's server management software, Intel® Server Control (ISC), Simple Network Management Protocol (SNMP) agents, and the Microsoft Windows* event log.   For example, Any problems that occur on a system are indicated by the LEDs on the alarms panel. ISC's PI (Platform Instrumentation) (PI) currently monitors and provides information on several baseboard and hardware components. Any exceeded thresholds or alerts that are received by the Local Response Agent (LRA) are captured forwarded as alarm requests to by TAM, which fields the errors alarm requests and illuminates corresponding status LEDs on the alarms panel. In addition to these hardware notifications, TAM can be configured to receive application and system events via SNMP. For SNMP traps, the SNMP event forwarder can be configured to map generated system, application, and operating system SNMP traps into alarm set and alarm clear requests to the TAM.

Configuration Requirements

Supported Hardware:

- HP cc3300 Carrier Grade Server and HP cc2300 Carrier Grade Server

Supported Software:

- Linux Red Hat 7.1*
- Intel Server Control
- Windows 2000 Advanced Server

Hardware Events

Any software agent that receives hardware-related events can send alarm panel requests to the Telco Alarms Manager. TAM provides a programming interface that allows integrated hardware vendors and integrated solutions integrators to develop additional event forwarders for their events.

The TAM software includes a pre-built event forwarder that receives hardware-related events from the Intel® Server Control software and sends appropriate TAM requests. The ISC integration is described in Section 5.

The TAM software also includes a pre-built SNMP event forwarder. This event forwarder may also receive hardware-related SNMP traps that can be translated and forwarded to the TAM as alarm requests. The TAM SNMP Event Forwarder is described in Section 5.

Software Events

Any software agent that receives software-related events can send alarm panel requests to the Telco Alarms Manager. TAM provides a programming interface that allows integrated solutions integrators to develop additional event forwarders for their software events.

The TAM software includes a pre-built SNMP event forwarder. This event forwarder may receive software-related SNMP traps that can be translated and forwarded to the TAM as alarm requests. The TAM SNMP Event Forwarder for Linux is described in Section 5.

The TAM software includes a pre-built event forwarder that processes Microsoft Windows* event log entries and sends appropriate TAM requests. The Telco Alarms Manager Event Forwarder also provides support for SNMP Trap filtering and event forwarding. The TAM Event Forwarder for Windows is described in Section 5.

TAM Event Forwarders

Intel Server Control Event Forwarding

Overview

TAM integrates with the Intel Server Control (ISC) product to provide hardware management information for use in setting an associated Telco alarm. If a hardware failure occurs, or a sensor threshold is exceeded, the PI receives the event information from the Baseboard Management Controller (BMC) via the Distributed Management Interface (DMI) service layer. The ISC Local Response Agent (LRA) monitors these events and translates the DMI event severity into the appropriate Telco Alarms Panel severity and forwards an alarm set request to the TAM.

Monitored Events

The TAM can activate LEDs when a problem arises in any one of the hardware areas monitored by ISC. Following is a summary of some of the hardware components monitored. Please refer to ISC documentation for more detailed information.

- Fan (failure, speed)
- Memory (single and multi bit errors, ECC errors)
- Processor (thermal trips and internal errors)
- Temperature (baseboard and processor temperature)
- Voltage (standby, baseboard, processors)
- Power supplies (presence, redundancy, temperature)

Configuration

No user intervention is required to receive hardware alerts. As long as ISC (Telco Version) is installed and running, hardware alert information is automatically passed to the alarms panel. Thresholds for hardware alerts can be configured via ISC's Platform Instrumentation Control.

DMI to TELCO Alarm Severity Mapping

ISC has five event severity classifications that map to the alarm panel's four LEDs. Table 1 shows the hardware event mapping. If the event regards power or voltage, the power LED will become active as well as the event's severity LED.

Table 1. Severity Mapping

ISC DMI Event Severity	TAM Alarm Panel Severity or Action	Comments
OK	Clear all alarm indications for this sensor	One or more previously reported alarms have been cleared.
INFORMATION	Clear all alarm indications for this sensor	Reporting of operation results.
NON-CRITICAL	MINOR (MNR)	A non-service-affecting condition. Corrective action should be taken in order to prevent a more serious fault.
CRITICAL	MAJOR (MJR)	A service-affecting condition that requires an urgent action.
NON-RECOVERABLE	CRITICAL (CRT)	A service-affecting condition that requires an immediate action.
	POWER (PWR)	Only active for power or voltage events.

TAM Software Alerts SNMP Event Forwarder for Linux Red Hat 7.1 and Red Hat Enterprise Linux AS 2.1

Overview

The SNMP agents provide a channel for application and system error forwarding to TAM.

Under Linux, the open source UCD Agent generates SNMP traps for events described by the operating system Management Information Block (MIB).   The UCD SNMP implementation also provides an SNMP trap daemon that allows filtering of traps and the execution of configured programs or scripts on receiving certain traps. Along with the TAM, Intel® provides a TAM SNMP Event Forwarder and its a corresponding configuration file that specifies how to translate and forward trap information as alarm requests to the Telco Alarm Manager. For example, when a threshold exceeded trap is generated with a trap event severity recognized by the TAM SNMP Event Forwarder, the event forwarder will send an alarm request to the TAM.

The TAM SNMP Event Forwarder can forward TAM alarm requests in the following ways:

- By filtering SNMP traps and sending TAM alarm requests for the associated problem severities

- By monitoring supported OIDs for value changes as configured in the configuration file, generating SNMP traps, and sending TAM alarm requests

This document does not go into details about using the UCD SNMP agent – it only describes what needs to be done to make the TAM work with the UCD agent. Please refer to the UCD documentation for more detailed information.   Also, note that the UCD Agent has been renamed in later revisions to the NET-SNMP Agent.

Trap Filtering Configuration

The following paragraphs describe the necessary configuration actions needed to set up TAM SNMP Event Forwarder trap filtering:

`snmptrapd.conf` – `snmptrapd.conf` is located in `/usr/share/snmp`. This file filters traps according to information in `snmptrapd.conf`. The `snmptrapd` daemon listens for traps and uses this file to determine what executable to launch and what parameters to pass into `stdin`. This file has been extended in order to provide information to enable the alarms panel to function according to specific trap severities that are encountered. The file has some commented information that can guide you through understanding the syntax. This file is also located in Appendix A of this document.

Following is the standard syntax for the `snmptrapd.conf` file and the syntax for the TAM SNMP Event Forwarder, `tamef`:

```
traphandle  trapOID  /path/executable  parameters
traphandle trapOID /usr/local/isc/tam/tamef sevOID ok min maj crit
```
- *sevOID*. is the SNMP OID used by *tamef* to obtain the severity of the current trap.

- *ok.* is a comma-delimited string that contains any severity values that indicate an "OK" status for the current trap. Traps received with any of these values causes the Event Forwarder to send a "clear" request to the TAM.

- *min.* is a comma-delimited string containing any values that indicate a minor status for the application trap.

- *maj* is a comma-delimited string containing any values that indicate a major status.

- *crit* is a comma-delimited string containing any values that indicate a critical status.

The configuration allows for mapping of any applications severity values to the Telco Alarms Panel's three categories of Minor, Major, and Critical.

Following is an example of how TAM interacts with snmptrapd.conf. To establish a trap severity-to-TAM alarm panel severity, the traphandle command is used.

For the example, application ABCApp has traps with the following severities:

Table 2: ABCApp Severity Level Definitions

ABCApp Trap Severity Level	Trap Keyword	Description	Selected TAM alarm panel severity mapping
0	emergencies	System unusable	critical
1	alerts	Immediate action required	critical
2	critical	Critical condition	major
3	errors	Error conditions	major
4	warning	Warning conditions	minor
5	notifications	Normal but significant conditions	minor
6	informational	Informational messages	OK
7	debugging	Debugging messages	<not mapped>

The standard `snmptrapd.conf` format is as follows:

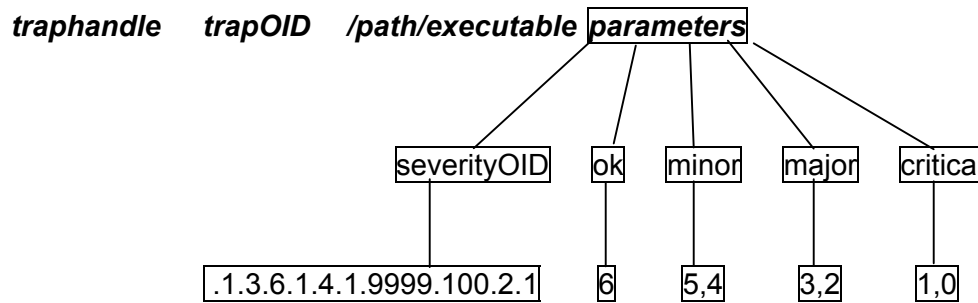***traphandle   trapOID   /path/executable parameters***

Figure 1: Traphandle

The resulting `tamef` entry is:

```
tTraphandle ABC-MIB::traps /usr/local/isc/tam/tamef .1.3.6.1.4.1.9999.2.1 6 5,4 3,2  1,0
```

Since the delivery of trap information is varied, severity, event IDs, and other trap information can be received a multitude of ways. In order for TAM to keep track of events, two things are essential. One is the severity and the other is a unique identifier for the event.

If the severity is sent as a SNMP variable instead of being set in the MIB (management information block), then you can replace the trapOID field with the following:

```
-P<snmpvar#>

traphandle ABC-MIB::traps /usr/local/isc/tam/tamef -P5 6 5,4 3,2 1,0
```

In this scenario, the fifth SNMP variable contains the trap severity.

OID Monitoring / Trap Generation Configuration

The following paragraphs describe the necessary configuration actions needed to set up TAMP SNMP Event Forwarder OID monitoring:

`tamep` – `tamep` is the TAM SNMP event poller. It constantly watches for changes in the OID values identified in `snmpd.conf`. Make sure that `$cs` is set to the community string.

`snmpd.conf` – `snmpd.conf` may be located in `/etc/snmp` or `/usr/share/snmp`. To ensure that the local system will receive its own trap notifications make certain that "`trapsink localhost`" is included in the file. Configure any thresholds for process checks, disk checks, and load average checks. Refer to commented-out text in `smpd.conf` and its main page for more information. By default, no thresholds are set in this file, so no system events will show on the alarms panel until this file is configured.

Monitored OIDs on Linux Red Hat *7.1 and Red Hat Enterprise Linux AS 2.1

The following is a list of OIDs monitored by default by the TAM SNMP Event Forwarder when used with the Linux Red Hat* distribution. Please refer to the UCD SNMP Agent and Red Hat OS MIB documentation for more information.

- prTable - A table containing information on running programs/daemons configured for monitoring in the SNMP.CONF of the agent. Processes violating the maximum number of running processes limit established by the agent's configuration file cause an alarm request to be forwarded to the alarms panel. This allows for alerting if a process dies or if too many instances are running.

- diskTable - Disk watching information. Partitions to be watched are configured by the SNMP.CONF of the agent. A minimum space in KB can be specified or a minimum percentage.

- loadTable - Load average information. If a 5, 10, or 15 minute average exceeds the configured maximum values, the alarms panel is notified.

- fileTable - Table of monitored files. Files are watched. If the maximum space in KB exceeds the configured value, the alarms panel is notified.

- memory - Monitors swap memory information. TAM is alerted if very little swap space is left.

- snmpErrs – Any trouble with the SNMP agent will notify the alarms panel.

TAM Event Forwarder for Windows* 2000 Advanced Server Configuration

Overview

Under Windows 2000 Advanced Server, the SNMP Trap Listener Service along with the Telco Alarms Manager Event Forwarder applications allow filtering of operating system events and SNMP traps that can be forwarded to TAM. The application provides a way to specify traps that the SNMP Trap Listener service can receive and it provides an interface to map application specific severities to the TAM LEDs.

The Telco Alarms Manager Event Forwarder application also uses the "Event to Trap Translator", a Windows 2000 tool that translates Windows 2000 system events into SNMP traps. Please refer to Windows 2000 documentation for more detailed information on using `evntwin.exe`.

Monitored Events

Telco Alarms Manager Event Forwarder (`tamef`) allows configuration for application- specified events and system events. In this application, the user can specify traps and a TAM severity that the trap indicates. The Event to Trap Translator application allows users to translate selected system events to traps which in turn allow these events to be forwarded to TAM.

Configuration

The executable for the Telco Alarms Manager is located in `%ISCPATH%\bin\tamef.exe`. Figure 2 below is an illustration of `tamef`.
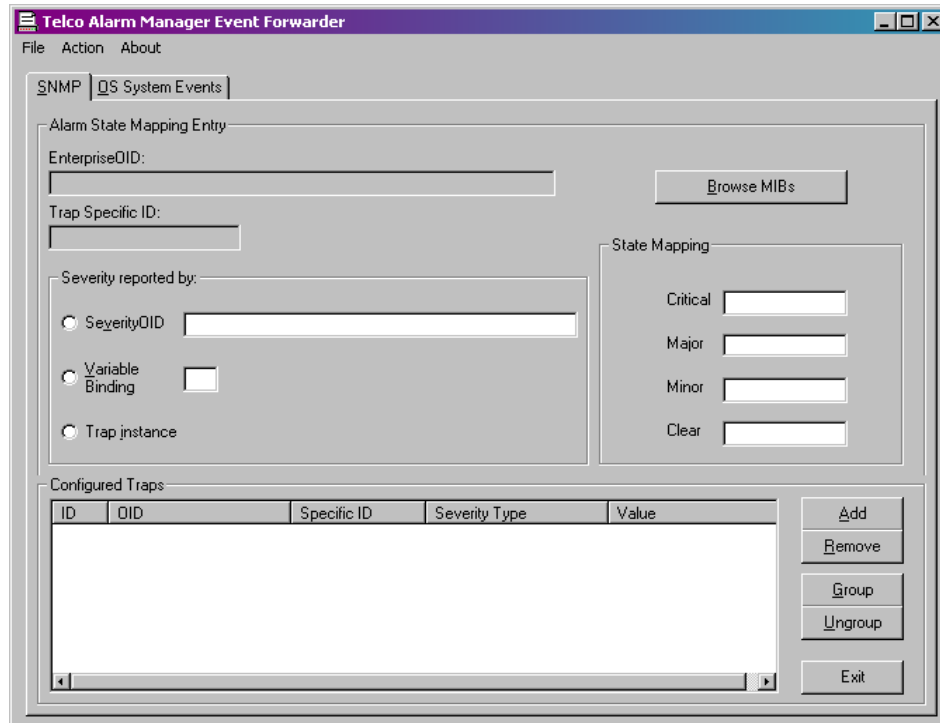

Figure 2 Telco Alarm Manager Event Forwarder

The SNMP tab allows you to:

- Enter SNMP Enterprise OID
- Enter SNMP trap identifiers
- Specify where a trap's severity is reported
- Map trap severity values to Telco Alarm Manager severity values.
- View configured traps
- Add, Delete, Group, and Ungroup configured traps

The "SeverityOID" text box allows you to specify an OID that reports the severity for a specific trap. If the severity is reported in the trap's variable bindings, you can enter a number indicating the nth variable to look at to obtain the severity of the trap. Selecting trap instance implies that the trap itself indicates a certain severity.

Once `tamef` knows how to find the severity, it needs to know how to map trap severities to the alarms panel LED severities. This is achieved by input in the state mappings section. In each field, enter a comma-separated list of trap severity values that indicate a TAM severity. For example, a trap reports 0 as

a 'back to normal' severity; 1,2, and 3 as minor severities; 4 and 5 as major severities; and 6 as a critical. Enter "0" in the clear text box, "1,2,3" in the minor text box, "4,5" in the major text box, and "6" in the critical.

Configured traps can be added or removed to the "Configured Traps" list box by clicking the Add or Remove buttons. Use the Group and Ungroup buttons to set configured traps to a common ID. This allows TAM to know what group an incoming trap message belongs to and whether to clear an LED or set the LED to a specific severity state. The configured traps can be grouped or ungrouped by selecting the trap in the "Configured Traps" list box, then click the "Group" or "Ungroup" button.

The "Event to Trap Translator" tab allows you to:

- Configure a trap based on system, application, and security system events

As stated earlier, Windows 2000 Advanced Server provides the "Event to Trap Translator" (evntwin.exe) application. System events are translated into enterprise OIDs and trap ids that can be used in the tamef application.   Please keep in mind that trap groupings in tamef must have a clearing event in order to report correct LED status. Even though evntwin has a column for severities, a lot of the severities are listed as informational or success; therefore, you should assign the evntwin-generated trap instance a severity within the tamef application.

Duplicating configured traps across multiple servers

The "Telco Alarms Manager Event Forwarder" generates a configuration file at location %ISCPATH%\bin\snmptraplistener.ini. The snmptraplistener service parses this file whenever a trap message is received.   Once created on one server, this file can be copied to other ISC servers that you wish to have identical tamef configurations.

Appendix A: SNMPTRAP.CONF

```
#*TAM*
# This section is designed for the Intel Telco Alarms
# Manager.  Below is an explanation of how this file is
# configured and some examples are provided.
#
# snmptrapd.conf accepts the following format:
#
#  traphandle OID command args
#
# For example:
#
#  traphandle .1.3.6.1.6.3.1.1.5.6  /home/nba/bin/traps  egp
#  traphandle SNMPv2-MIB::coldStart /home/nba/bin/traps  cold
#  traphandle SNMPv2-MIB::warmStart   /home/nba/bin/traps  warm
#  traphandle IF-MIB::linkDown        /home/nba/bin/traps  down
#
# For traps to be received and fielded to the Telco Alarms Panel, please
# use the following arguments:
#
#  traphandle OID command severityOID/parameter ok minor major critcal
#
# where
# severityOID/parameter is the OID that stores the integer value
representing the
#   application's severity or –Pn, where n is the nth argument that is
the severity,
# critcal which is a comma delimited list of numbers that will cause the
#   critical LED to illuminate,
# major which is a comma delimited list of numbers that will cause the
major
#   LED to illuminate, and
# minor which is a comma delimited list of numbers that will cause the
minor
#   LED to illuminate.
#
# For example, application Foo has it's trap severity information stored
in
# .1.3.6.1.4.1.9999.1.2.3.4.0 and it has 6 severities:  1 is
information,
# 2 is minor, 3 is warning 4 is critical 5 is severe and 6 is non-
recoverable.
# If the application doesn't have a specific severity, such as
"warning", then
# place an "X" in that field.
#
#   traphandle  .1.3.6.1.4.1.9999.10.20.3  .1.3.6.1.4.1.9999.1.2.3.4.0
0,1 2 3,4 5,6
#

traphandle .1.3.6.1.4.1.ucdavis.prTable.0.2021 /usr/local/isc/tam/tamef
.1.3.6.1.4.1.2021.2.1.100.1 0 X 1 X
```

```
traphandle .1.3.6.1.4.1.ucdavis.memory.0.2021 /usr/local/isc/tam/tamef
.1.3.6.1.4.1.2021.4.100.0 0 X 1 X
traphandle .1.3.6.1.4.1.ucdavis.dskTable.0.2021 /usr/local/isc/tam/tamef
.1.3.6.1.4.1.2021.9.100.1 0 X 1 X
traphandle .1.3.6.1.4.1.ucdavis.laTable.0.2021 /usr/local/isc/tam/tamef
.1.3.6.1.4.1.2021.10.100.1 0 X 1 X
traphandle .1.3.6.1.4.1.ucdavis.fileTable.0.2021
/usr/local/isc/tam/tamef .1.3.6.1.4.1.2021.15.100.1 0 X 1 X
traphandle .1.3.6.1.4.1.ucdavis.snmperrs.0.2021 /usr/local/isc/tam/tamef
.1.3.6.1.4.1.2021.101.100.1 0 X 1 X
#*TAM*
```